www.icgst.com

Multidimensional Approach for Securing Images on Cloud

D. Boopathy[1], M. Sundaresan[2]
[1 & 2]*Department of Information Technology, Bharathiar University, Coimbatore- 641046,Tamilnadu, India*
ndboopathy@gmail.com

Abstract
Encryption is one of the methodologies used to maintain and protect the data confidentiality. As per the user data type's requirements, users need to adopt and implement any one of the existing methods. But those encryption methods and standards may not be bound within the user data country regulations, when the users are from different geographical locations. Some of the existing methods are already compromised by hackers and also some of the government agencies are forcing their country based service providers to provide the encrypted information in the name to maintain the country's security. It is very difficult to manage the threats with one method. The proposed method tried its maximum level to reduce the threats by using different points of view. In this proposed method images and the block-based encryption method have been used to protect the normal and sensitive image from the unauthorized access. The proposed method is tested on all proposed encryption types using greyscale in two scenarios. They are Different Images One Type (DIOT) and Single Image All Types (SIAT). The results of the proposed methods are evaluated using PSNR, MSE, Size of the Image and Histogram to verify the image's integrity.[1]

Keywords: *Image Encryption, Decryption, Image Security, Greyscale Images, Cloud Security.*

Nomenclature

SCDSPM	Secured Cloud Data Storage Prototype Model
E & DGM	Encryption & Decryption Gateway Model
MDE & DPM	Multi-Dimensional Encryption & Decryption Model
PRA	Pixel Rearrange Algorithm
PRRA	Pixel Reverse Rearrange Algorithm
PSA	Pixel Shuffling Algorithm
PRSA	Pixel Reverse Shuffling Algorithm
DIOT	Different Images One Type
SIAT	Single Image All Types

[1]This study has been implemented and Tested on Java platform at Department of Information Technology, Bharathiar University, Coimbatore, Tamilnadu, India.

1. Introduction
"One picture is worth a thousand words", is a popular English saying which has been used since 1918, in a newspaper advertisement for the San Antonio Light [1]. The image conveys the complete information to the viewers without any loss of any piece of information. Sensitive images must be safeguarded from the general viewers and unauthorized viewers in order to protect the confidential nature of its contents. For this purpose, different encryption standards are applied on the sensitive and non-sensitive images to maintain the image confidentiality and to prevent that image from being mishandled by the unauthorized and unidentified users. While coming to the specific objective, the existing encryption standards in use are not reliable due to their limitations, data processing technique and algorithm working architecture methods. Once the images are stored online, then the owner of the images automatically loses his rights on those images. The online service providers are altering their policies in data handling and even reformatted the policy related data from time to time without any users' interactions. That service provider's server may be geographically positioned in some other vicinity and in that place only the encryption and decryption will take place. Once the user encrypts the data by using a specific service provider, then the user needs to decrypt that data by using that same service provider only but it may be done from anywhere because they are online. If the user is using the offline encryption tools, then the user needs to depend on that device for the encryption and decryption, but the user must always keep the device with him to perform either encryption or decryption whenever necessary. Taking all these things into consideration, the Secured Cloud Data Storage Prototype Model is designed and the Multi-Dimensional Encryption and Decryption Method is one of the modules in that. Section 2 reviews the related works concerning the encryption techniques to maintain the security of the data storage. Section 3 deliberates on the different working methodology, procedure, Pseudo code and Testing file details of MDE & DPM Algorithm. Section 4 delineates the implementation, and Section 5 explains the experimental results and in addition the features of the proposed method are also discussed here. Section 6 presents the conclusion derived from the findings, the

advantages of the proposed algorithm and finally its related future enhancements.

2. Literature Review

New image encryption design which utilizes one of the three dynamic chaotic [2] systems to shuffle the location of the image pixels and uses another one of the same three chaotic maps to mystify the association between the cipher image and the plain-image, thereby considerably increasing the resistance to attacks. To overcome this, Sakthidasan et al proposed the algorithm with the advantage of bigger key space, lesser iteration times and high security analysis such as key space analysis, statistical analysis and sensitivity analysis [3].Navitha et al proposed a very new and combined approach for DCT based image compression, pixel shuffling based encryption, decryption and steganography for real-time applications [4]. Quist et al suggested the sets out method to contribute to the general body of knowledge in the area of cryptography application by developing a cipher algorithm for image encryption of m*n size by shuffling the RGB pixel values. The algorithm ultimately makes it possible for encryption and decryption of the images based on the RGB pixel [5]. Junqin et al introduced a permutation-substitution image encryption scheme based on generalized Arnold map. Only one round of permutation and one round of substitution are performed to get the desirable results. The generalized chaotic Arnold maps are applied to generate the pseudo-random sequences for the permutation and substitution [6]. Lohit et al explored the implementation of AES in MATLAB on plaintext encryption and cipher text decryption. These results are superior to the similar software implementations of AES [7].

3. Methodology

The existing methods, updated algorithms are using different concepts and implementations of these are enough to handle the data encryption process in offline mode but not in online mode. In online mode, i.e.in cloud [8], the existing methods require more time and utilize more resources to perform the encryption and decryption process. The geographically distributed data processing servers will raise the security breach issues and data trans-border related issues. So, the data need to be encrypted before the data are transferred from the user end to the server end. The proposed method considered all of these measures and provides the prototype model with different modules to overcome the data related storage, retrieval and encryption issues.

SCDSPM

The Secured Cloud Data Storage Prototype Model [9, 10] contains four sub-modules; they are:

- Authentication Authorization Resolving Module [11, 12]
- Data Type Identification and Extension Validation Module [13]
- Encryption and Decryption Gateway Module [14 - 16]
- Automatic Cloud Data Backup Module [17]

This paper explains the third module of SCDSPM i.e. E & DGM. This E & DGM is redefined with some modification and named in this paper as Multi-Dimensional Encryption and Decryption Module (MDE & DPM). Figure 1 shows the proposed Multi-Dimensional Encryption and Decryption Module (MDE & DPM).

Multi-Dimensional Encryption and Decryption Module framework

Using the new type of encryption method will avoid the user's data from superfluous risks. Each and every encryption and decryption logics must be uniquely different from other methods. In that way, the proposed encryption algorithm is using new logic and it will help to avoid the unconstitutional access, illicit usage and unlawful surveillance of the user's data by unauthorized persons.

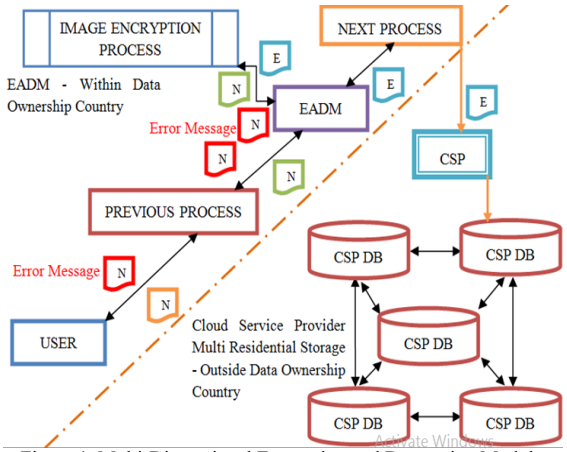


Figure 1. Multi-Dimensional Encryption and Decryption Module

The proposed Multi-Dimensional Encryption and Decryption Module presently concentrated on image format files only. This paper explains the proposed Multi-Dimensional Encryption and Decryption Module with tested standard and non-standard images and its related experimental results. It uses 512 x 512 pixel [18] images for testing purposes. Multi-Dimensional Encryption and Decryption Module contains four different algorithms to encrypt and decrypt the image. The four algorithms are: 1. Pixel Rearrange Algorithm, 2. Pixel Shuffling Algorithm, 3. Pixel Reverse Rearrange Algorithm, 4. Pixel Reverse Shuffling Algorithm.

The above mentioned algorithms are tested with different test case images which include standard images and non-standard images. Figures 2(a) and 2(b) shows the MDE & DPM's encryption and decryption method.

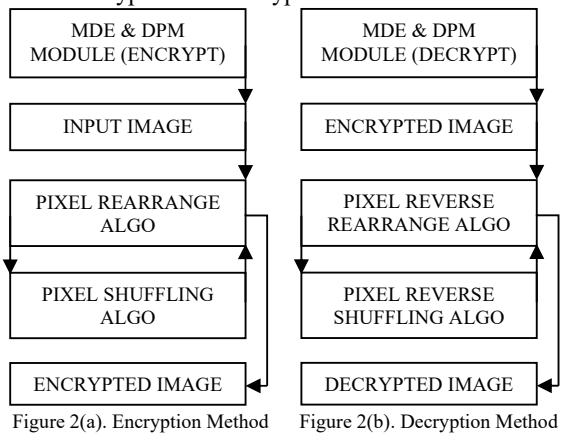


Figure 2(a). Encryption Method

Figure 2(b). Decryption Method

Pixel Rearrange Algorithm (PRA)

In Pixel Rearrange Algorithm the image pixels are rearranged into different positions using the 4 X 4 matrix concept. The pixel values of the images are relocated to other positions from their original positions. Once the image pixels are relocated to another position, then they automatically reflect in the original structural content of the image. Pixel Rearrange Algorithm is holding 4096 possible ways to rearrange the image pixels into a new position within the selected 4 X 4 matrix method. The result obtained from PRA is incorporated into the PSA.

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

1	9	10	8	R_n
16	2	7	11	R_n
14	6	3	12	R_n
5	15	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 3. Before applying PRA and after applying PRA

Figure3 shows the image pixel location before applying Pixel Rearrange Algorithm (PRA) and also shows the image pixel location after applying Pixel Rearrange Algorithm (PRA).

Pixel Reverse Rearrange Algorithm (PRRA)

The PRRA algorithm is used to reverse the Pixel Rearrange Algorithm's (PRA) relocated pixel values into their original position i.e. original location. The reversing method will use the rearrange method information from the decryption key.

1	9	10	8	R_n
16	2	7	11	R_n
14	6	3	12	R_n
5	15	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 4. Before applying PRRA and after applying PRRA

Figure 4 shows the pixel location before applying Pixel Reverse Rearrange Algorithm (PRRA) and also shows the pixel location after applying Pixel Reverse Rearrange Algorithm (PRRA).

Pixel Shuffling Algorithm (PSA)

The Pixel Shuffling Algorithm (PSA) is used to shuffle the pixel values within the matrix value. This research work holds sixteen different types of pixel values shuffling methods. Within those different methods, one of the methods will be automatically (i.e. randomly) selected and applied by the Pixel Shuffling Algorithm (PSA); then the selected method results will be stored with the decryption key. In each and every pixel shuffling method, one of the value locations will be fixed as a constant to identify which shuffling method is used to shuffle the pixel values. Here the decryption key will be automatically generated by the PSA algorithm with PSA related information and that information will be used at the time of decryption.

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

9	1	10	8	R_n
16	2	11	7	R_n
15	6	12	3	R_n
5	14	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 5. Before applying PSA and after applying PSA

Figure5 shows the pixel location before applying Pixel Shuffling Algorithm (PSA) and also shows the pixel location after applying Pixel Shuffling Algorithm (PSA). In the selected pixel shuffling method, the pixel value 14 is fixed as a constant value to identify the shuffling method.

Pixel Reverse Shuffling Algorithm (PRSA)

The decryption key holds the used Pixel shuffling algorithm's information. By using that information only the pixel reverse shuffling algorithm will work. Once the Pixel Reverse Shuffling Algorithm (PRSA) gets the information from the decryption key, then it will apply that correlated reverse shuffling method on that shuffled image pixel values. Once the pixel values are reversed, then it needs to be processed with the Pixel Reverse Rearrange Algorithm (PRRA). Then only the original structured content of the image will be constructed.

9	1	10	8	R_n
16	2	11	7	R_n
15	6	12	3	R_n
5	14	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 6. Before applying PRSA and after applying PRSA

Figure 6 shows the pixel location before applying Pixel Reverse Shuffling Algorithm (PRSA) and also shows the pixel location after applying Pixel Reverse Shuffling Algorithm (PRSA). By using the pixel value 14, which is fixed as constant value, is used to identify the shuffling method.

The Pixel Rearrange Algorithm (PRA) and Pixel Shuffling Algorithm (PSA) are used to encrypt the image. The Pixel Reverse Rearrange Algorithm (PRRA) and Pixel Reverse Shuffling Algorithm (PRSA) are used to decrypt the image.

Pseudo code for MDE&DMF

Encryption pseudo code:

Get the image from the user
Store that image into an **Object**
Read the Object Pixel Values
Store that Object Pixel Values into a Red, Green and Blue band color Text File
Get the Pixel Values from that Text Files
Store that Pixel Values of Text Files as three **Objects**
Apply the **PRA** on all the **Objects**
Apply the **PSA** on all the **Objects**
Apply the **PRA** on all the **Objects**
Prepare the **Decryption Key** with used algorithm method information
Convert all the Pixel Values Text Files and apply respective color band and merge all that color band files into an **Image File**
Store that **Image File** into selected storage in selected format
Store that **Image Decryption Key** into the selected storage in desired format

Decryption pseudo code:

Get the image from the user
Store that image into an **Object**

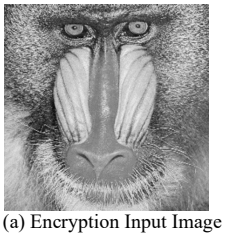
Get the Decryption Key to apply and decrypt the image
If the key got authenticated **Then**
Forward the process to next step
Else
Show an error message as key is invalid and **STOP** the process
Read the Object Pixel Values
Store that Object Pixel Values a Red, Green and Blue band color Text File
Get the Pixel Values from that Text Files
Store that Pixel Values of Text Files as three **Objects**
Apply the **PRRA** on all the **Objects**
Apply the **PRSA** on all the **Objects**
Apply the **PRRA** on all the **Objects**
Convert all the Pixel Values Text Files and apply respective color band and merge all that color band files into an **Image File**
Store that **Image File** into the selected storage in selected format

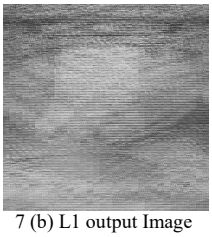
4. Implementation

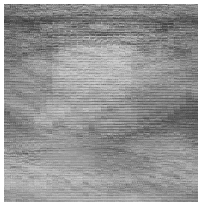
The proposed method has been implemented using the MATLAB simulation tool and Java 1.8 programming tool. The implementation is divided into two parts. The first part is to read and write the three band image pixel values into the text file and then it needs to read the three band image pixel values from the text file and to construct the image file. Five different standard images [19] and two non-standard images are taken for testing purpose. Each testing file contains 512 x 512 pixel image. The remaining details of the testing images are shown in Table 1.

Table 1. Testing Images

Image Sl. No.	Image Name	Standard / Normal Image	Image Size
1	Baboon	Standard Image / Tiff Format	258 KB
2	Cameraman	Standard Image / Tiff Format	256 KB
3	Lena	Standard Image / Tiff Format	260 KB
4	Pirate	Non-Standard Image / Tiff Format	257 KB
5	Room	Non-Standard Image / Tiff Format	258 KB
6	Peppers	Standard Image / Tiff Format	206 KB
7	House	Standard Image / Tiff Format	106 KB


7 (a) Encryption Input Image


7 (b) L1 output Image


7 (c) L2 output Image

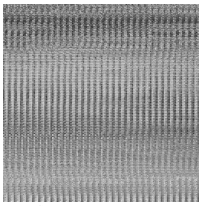
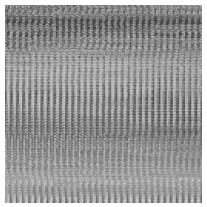
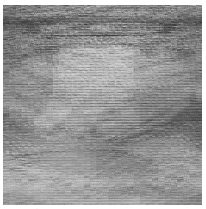
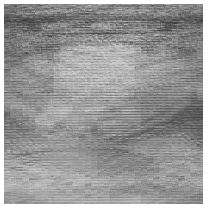

7 (d) L3 output Image

Figure 7 (a - d). Step by Step Encryption Process


8 (a) Decryption Input Image


8 (b) L1 output Image


8 (c) L2 output Image

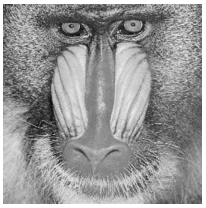

8 (d) L3 output Image

Figure 8 (a - d). Step by Step Decryption Process

The step by step working formation on image of the proposed encryption and decryption algorithm is applied on the baboon standard "TIFF" [20] image file format [21] and the resultant images are shown above. Figure 7(a) is the input image, figure 7(b) is the first level output image, figure 7(c) is the second level output image, and figure 7(d) is the final level output image i.e., encrypted image. Similarly figure 8(a) is the encrypted image, figure 8(b) is the first level output image, figure 8(c) is the second level output image, and figure 8(d) is the final level output image i.e. decrypted image. There are sixteen different types of pixel shuffling algorithms are available in the proposed Pixel Shuffling algorithm. Among those pixel shuffling methods, for testing purpose all the algorithms are used in this paper.

5. Results and Discussion

There is no change found on the histogram of normal image, encrypted image and decrypted image. The histogram of Baboon image for normal image, encrypted image and decrypted image is shown below in figure 9 (a-c). The implementation is done in two ways, they are:

- Different Image One Type method (DIOT)
- Single Image All Types method (SIAT)

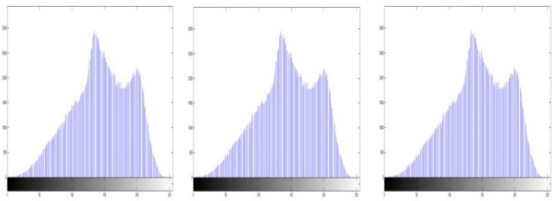

9 (a) 9 (b) 9 (c)

Figure 9 (a - c). Histogram of Normal, Encrypted and Decrypted Image

Figure 9(a) shows the normal input image histogram [22], Figure 9(b) shows the encrypted image histogram [22] and Figure 9(c) shows the decrypted image histogram [22]. Table 2 shows 7 different Input images and its related encrypted image and decrypted image in Different Images One Type method (DIOT). The different images are processed in one of the proposed encryption algorithms to verify the algorithm working style. In that, Type-16 encryption algorithm has been used to encrypt the test images. In the table 2, the 1a – 7a images are

normal input image, the 1b – 7b are encrypted image and 1c – 7c are decrypted image.

Table 2. DIOT related Normal, Encrypted and Decrypted Image

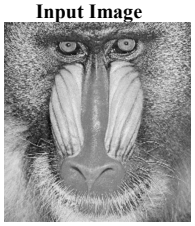
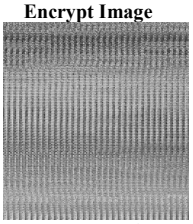
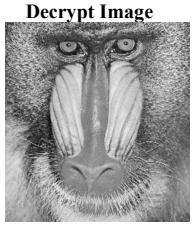
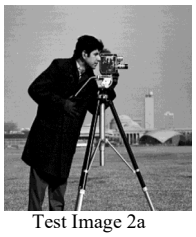
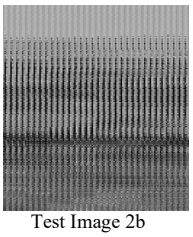
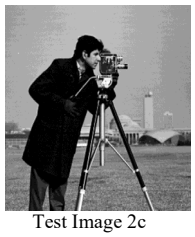
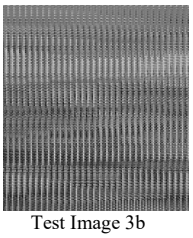
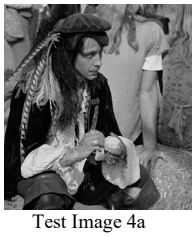
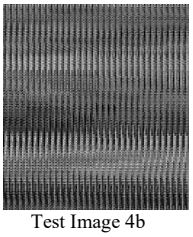
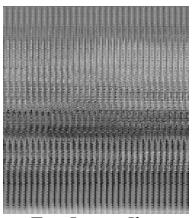
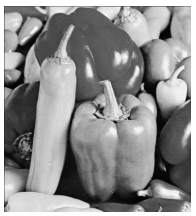
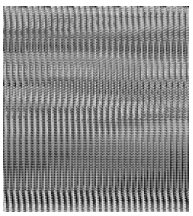
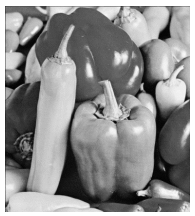
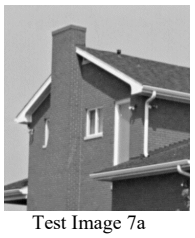
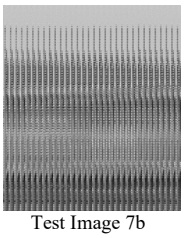
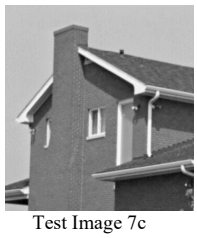
Input Image	Encrypt Image	Decrypt Image
 Test Image 1a	 Test Image 1b	 Test Image 1c
 Test Image 2a	 Test Image 2b	 Test Image 2c
 Test Image 3a	 Test Image 3b	 Test Image 3c
 Test Image 4a	 Test Image 4b	 Test Image 4c
 Test Image 5a	 Test Image 5b	 Test Image 5c
 Test Image 6a	 Test Image 6b	 Test Image 6c
 Test Image 7a	 Test Image 7b	 Test Image 7c

Table 4 shows the parameters used to verify the comparison between encryption image and decryption image with normal image in DIOT. The Size of the Image, isequal() Function [23], PSNR [24] and MSE [25] are taken as parameters and they are compared among input image, encrypted image and decrypted image.

Table 3 shows the input image and the proposed encryption algorithms (i.e. From Type-01 to Type-16) encrypted images in Single Image All Types method i.e. SIAT. All the encrypted images might look like the same, but the differences are present in the Type-01 to Type-16 encrypted image outputs. The parameters used to measure the differences between the encrypted images are shown in table 5. The Single image is processed in all the proposed encryption algorithms, i.e. Type-01 to Type-16 to verify that all the algorithm's encrypted images are different from one another or not.

Table 3. SIAT related Normal, Encrypted and Decrypted Image

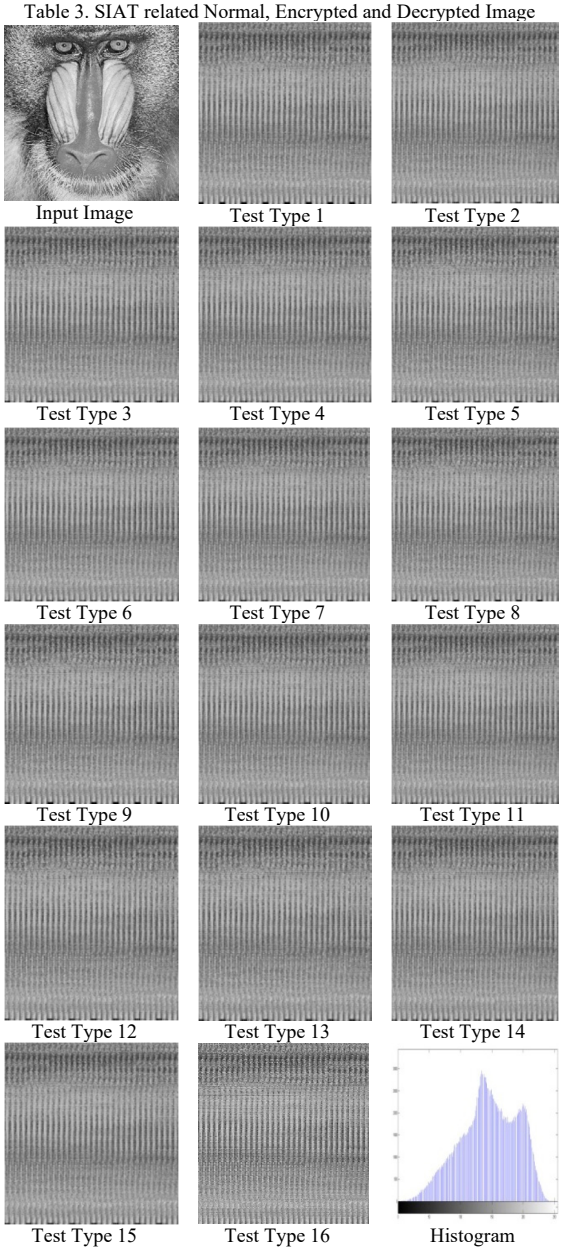


Table 4. Comparison of size of the image, isequal() Function, PSNR and MSE for Different Images One Type (DIOT)

Image Name	Details		Size of the Image	isequal ( ) Function	PSNR Value	MSE Rate
Test Image 1	Input Test Image Vs	Encryption Image	258 kB	0	33.8481328 dB	108.08
		Decryption Image	258 kB	1	Inf dB	0
Test Image 2	Input Test Image Vs	Encryption Image	256 kB	0	34.7574709 dB	87.66
		Decryption Image	256 kB	1	Inf dB	0
Test Image 3	Input Test Image Vs	Encryption Image	260 kB	0	33.7661045 dB	110.14
		Decryption Image	260 kB	1	Inf dB	0
Test Image 4	Input Test Image Vs	Encryption Image	257 kB	0	33.7901752 dB	109.53
		Decryption Image	257 kB	1	Inf dB	0
Test Image 5	Input Test Image Vs	Encryption Image	258 kB	0	34.0635209 dB	102.85
		Decryption Image	258 kB	1	Inf dB	0
Test Image 6	Input Test Image Vs	Encryption Image	206 kB	0	35.1889686 dB	79.37
		Decryption Image	206 kB	1	Inf dB	0
Test Image 7	Input Test Image Vs	Encryption Image	106 kB	0	33.7953091 dB	109.40
		Decryption Image	106 kB	1	Inf dB	0

Table 5. Comparison of size of the image, isequal() Function, PSNR and MSE for Single Image All Types (SIAT)

Algorithm Type	Details		Size of the Image	isequal ( ) Function	PSNR Value	MSE Rate
Type 1	Input Test Image Vs	Encryption Image	234 kB	0	27.7668073 dB	108.74
		Decryption Image	234 kB	1	Inf dB	0
Type 2	Input Test Image Vs	Encryption Image	234 kB	0	27.7640038 dB	108.81
		Decryption Image	234 kB	1	Inf dB	0
Type 3	Input Test Image Vs	Encryption Image	234 kB	0	27.7620891 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 4	Input Test Image Vs	Encryption Image	234 kB	0	27.7621153 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 5	Input Test Image Vs	Encryption Image	234 kB	0	27.7555824 dB	109.02
		Decryption Image	234 kB	1	Inf dB	0
Type 6	Input Test Image Vs	Encryption Image	234 kB	0	27.7684094 dB	108.70
		Decryption Image	234 kB	1	Inf dB	0
Type 7	Input Test Image Vs	Encryption Image	234 kB	0	27.7640273 dB	108.81
		Decryption Image	234 kB	1	Inf dB	0
Type 8	Input Test Image Vs	Encryption Image	234 kB	0	27.7596387 dB	108.92
		Decryption Image	234 kB	1	Inf dB	0
Type 9	Input Test Image Vs	Encryption Image	234 kB	0	27.7571000 dB	108.99
		Decryption Image	234 kB	1	Inf dB	0
Type 10	Input Test Image Vs	Encryption Image	234 kB	0	27.7681899 dB	108.71
		Decryption Image	234 kB	1	Inf dB	0
Type 11	Input Test Image Vs	Encryption Image	234 kB	0	27.7622032 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 12	Input Test Image Vs	Encryption Image	234 kB	0	27.7632227 dB	108.83
		Decryption Image	234 kB	1	Inf dB	0
Type 13	Input Test Image Vs	Encryption Image	234 kB	0	27.7643787 dB	108.80
		Decryption Image	234 kB	1	Inf dB	0
Type 14	Input Test Image Vs	Encryption Image	234 kB	0	27.7658733 dB	108.77
		Decryption Image	234 kB	1	Inf dB	0
Type 15	Input Test Image Vs	Encryption Image	234 kB	0	27.7622326 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 16	Input Test Image Vs	Encryption Image	234 kB	0	27.7600431 dB	108.91
		Decryption Image	234 kB	1	Inf dB	0


The Size of the Image, isequal() Function, PSNR and MSE are taken as parameters and those things are compared among the input image, encrypted image and the decrypted image in DIOT and SIAT. The above mentioned parameters have been proved that the proposed algorithms are encrypted and decrypted the testing images and also the same parameters are proving that, there are differences presented between the sixteen different encryption algorithm's encrypted images in SIAT. The PSNR Value and MSE Rate equations used for calculations are shown below.

$$\frac{\sum M,N\left[I_2(m,n)-I_2(m,n)\right]^2}{M*N} \quad (1)$$

$$10\log_{10}\left(\frac{R^2}{MSE}\right) \quad (2)$$

The Equation 1 is used to calculate the MSE Rate and the Equation 2 is used to calculate the PSNR Value.

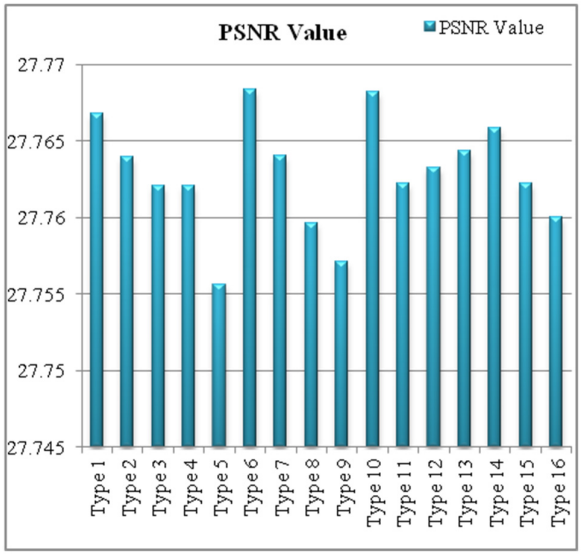


Figure 10. PSNR Differences between 16 Algorithms' Output

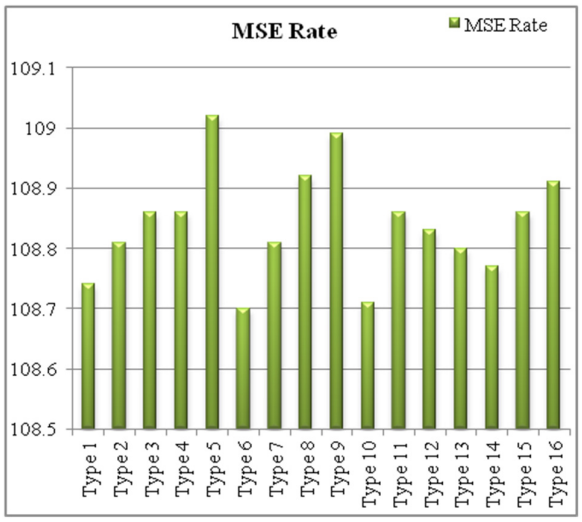


Figure 11. MSE Differences between 16 Algorithms' Output

Figure 10 shows the differences in PSNR value among the sixteen different proposed encryption algorithms. Figure 11 shows the differences in MSE rate among the sixteen different proposed encryption algorithms. The figure 10 and figure 11 details proven that, there are differences presented among the sixteen proposed algorithms and the encrypted images are different from one another algorithm's output.

6. Conclusion

The users' rights and privacy should not to be affected in online image encryption, and the users' image need to be encrypted within the users' country border limit. Moreover, the encrypted images needed to be kept back within users' country border limits and finally the users and their service provider who need to maintain the trust between them are the most important highlighted requirements of the cloud service users. Those things are covered in the Secured Cloud Data Storage Prototype Model and in that proposed model the image's confidentiality will be taken care of by Multi-Dimensional Encryption and Decryption Model. The proposed method has encrypted and decrypted the test images successfully in both multiple image single method testing and single image all propose algorithm methods testing. The single image all proposed algorithm methods testing has proved that all the 16 different encryption methods are not the same; each and every method will provide different encrypted image. The algorithm testing has been done on 512 x 512 pixel images only. In future the proposed algorithm will be extended to handle any size of images for encryption and decryption process to maintain the confidentiality of the image in online.

7. References

[1] https://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words, Date of Accessed: 09/08/2017

[2] D. Desai1, A. Prasad, J. Crasto, Chaos-Based System for Image Encryption, International Journal of Computer Science and Information Technologies, Vol. 3 , No. 4, pp. 4809-4811, 2012.

[3] K. Sakthidasan and B. V. Santhosh Krishna, A New Chaotic Algorithm for Image Encryption and Decryption of Digital Color Images, International Journal of Information and Education Technology, Vol. 1, No. 2, pp. 137 – 141, 2011.

[4] N. Agarwal, H. Sharma, An Efficient Pixel-shuffling Based Approach to Simultaneously Perform Image Compression, Encryption and Steganography, IJCSMC, Vol. 2, No. 5, pp. 376 – 385, 2013.

[5] Q-A Kester, Image Encryption based on the RGB PIXEL Transposition and Shuffling, International Journal of Computer Network and Information Security, Vol. 7, pp. 43-50, 2013.

[6] J. Zhao, W. Guo, R. Ye, A Chaos-based Image Encryption Scheme Using Permutation-Substitution Architecture, International Journal of Computer Trends and Technology, Vol. 15 No. 4, pp. 174 – 185, 2014.

[7] D. Lohit Kumar, Dr. A.R.Reddy, Dr.S.A.K.Jilani, Implementation of 128-bit AES algorithm in MATLAB", International Journal of Engineering Trends and Technology, Vol. 33 No. 3, pp. 126 – 129, 2016.
[8] http://www.oxforddictionaries.com/definition/english/cloud-computing, Date of Accessed: 05/05/2017.
[9] D. Boopathy and Dr. M. Sundaresan, Securing Public Data Storage in Cloud Environment, ICT and Critical Infrastructure: 48th Annual Convention of Computer Society of India, Visakhapatnam, India, pp.555-562, 2013.
[10] D. Boopathy and Dr. M. Sundaresan, Secured Cloud Data Storage – Prototype Trust Model for Public Cloud Storage, International Conference on Information and Communication Technology for Sustainable Development, Ahmadabad, India, pp.329–337, 2015.
[11] D. Boopathy and Dr. M. Sundaresan, Framework Model and Algorithm of Request based One Time Passkey (ROTP) Mechanism to Authenticate Cloud Users in Secured Way, 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, pp.5317–5322, 2016.
[12] D. Boopathy and Dr. M. Sundaresan, A Framework for User Authentication and Authorization using Request based One Time Passkey and User Active Session Identification, International Journal of Computer applications, Vol.172, No.10, pp.18–23, 2017.
[13] D. Boopathy and Dr. M. Sundaresan, Data Type Identification and Extension Validator Framework Model for Public Cloud Storage, Big Data Analytics - Proceedings of the 50th Annual Convention of Computer Society of India, New Delhi, India, pp.533–541, 2014.
[14] D. Boopathy and Dr. M. Sundaresan, Data Encryption Framework Model with Watermark Security for Data Storage in Public Cloud Model, IEEE Eighth International Conference on Computing for Sustainable Global Development, New Delhi, India, pp.1040–1044, 2014.
[15] D. Boopathy and Dr. M. Sundaresan, Enhanced Encryption and Decryption Gateway Model for Cloud Data Security in Cloud Storage, Emerging ICT for Bridging the Future - 49th Annual Convention of Computer Society of India, Hyderabad, India, pp.415–421, 2014.
[16] D. Boopathy and Dr. M. Sundaresan, Policy Based Data Encryption Mechanism Framework Model for Data Storage in Public Cloud Service Deployment Model, Elsevier Fourth International Joint Conference on Advances in Computer Science , Haryana, India, pp.423–429, 2013.
[17] D. Boopathy and Dr. M. Sundaresan, IDOCA and ODOCA – Enhanced Technique for Secured Cloud Data Storage, International Journal of Intelligent Engineering and Systems, Vol.10, No.06, pp. 49 - 59, 2017.
[18] https://en.wikipedia.org/wiki/Pixel, Date of Accessed: 05/11/2017.
[19] https://en.wikipedia.org/wiki/Standard_test_image, Date of Accessed: 05/11/2017.
[20] https://en.wikipedia.org/wiki/TIFF, Date of Accessed: 11/11/2017.
[21] https://en.wikipedia.org/wiki/Image_file_formats, Date of Accessed: 11/11/2017.
[22] http://in.mathworks.com/help/matlab/ref/isequal.html, Date of Accessed: 18/11/2017.
[23] https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio,Date of Accessed: 18/11/2017.
[24] https://en.wikipedia.org/wiki/Mean_squared_error, Date of Accessed: 18/11/2017.
[25] https://en.wikipedia.org/wiki/Histogram, Date of Accessed: 20/11/2017.

Biographies

D.Boopathy is a Research Scholar doing his PhD in Computer Science in the Department of Information Technology at Bharathiar University. He is qualified with M.Sc.(IT) and MCA from Bharathiar University. He did his Master of Philosophy in Computer Science at Dr. G.R.D College of Science, Coimbatore. His areas of interests are Information Security, Data Privacy and Cloud Computing. He is a Life Member of Computer Society of India and Indian Science Congress Association. So far he has co-authored 2 book chapters for 2 edited books (2 for IGI Global USA).
Email ID: ndboopathy@gmail.com

Prof.Dr.M.Sundaresan is currently Professor and Head of the Department of Information Technology at Bharathiar University, Coimbatore, India. He holds PhD in computer science. He has contributed more than 50 research papers in different areas of Computer Science such as Image Processing, Data Compression, Natural Language Processing, Speech Processing and Cloud Computing in reputed journals. He is a Senior and Life Member of Professional Bodies such as Computer Society of India, Indian Science Congress Association, and Indian Society for Technical Education and IACSIT. He is also in editorial board of five journals. He is the Sectional President for Information and Communication Science & Technology (including Computer Sciences) section in 105th Indian Science Congress. He is the Regional Vice President for Regional VII in Computer Society of India. So far he has authored 2 book chapters for 2 edited books (2 for IGI Global USA). Email ID: bu.sundaresan@gmail.com